ELSEVIER

Artificial Intelligence 74 (1995) 351-365

Artificial Intelligence

Research Note

Nested abnormality theories

Vladimir Lifschitz *

Department of Computer Sciences and Department of Philosophy, University of Texas at Austin, Austin, TX 78712, USA

Received April 1994; revised February 1995

Abstract

We propose a new approach to the use of circumscription for representing knowledge. Nested abnormality theories are similar to simple abnormality theories introduced by McCarthy, except that their axioms may have a nested structure, with each level corresponding to another application of the circumscription operator. The new style of applying circumscription sometimes leads to more economical and elegant formalizations. Mathematical properties of nested abnormality theories may be easier to investigate. These advantages are demonstrated by recasting several familiar applications of circumscription in the new format, including some examples of inheritance hierarchies, the domain closure assumption and causal minimization. Nested abnormality theories provide also a convenient representation for the explanation closure approach to the frame problem developed by Schubert.

1. Introduction

The methodology for representing defaults developed by McCarthy [15] involves the use of an "abnormality predicate" and the application of circumscription to minimize its extent. Since circumscribing abnormality can be performed in many different ways, one needs to decide which strategy to follow. McCarthy explored several possibilities, and none of them turned out to be completely satisfactory.

One of his proposals is to use *simple abnormality theories*, in which the circumscription of abnormality is done with all predicates varied. Because simple abnormality theories employ a standard circumscription policy, such a theory is completely characterized by the list of its axioms $A_1, \ldots, A_n$, just like an axiomatic theory in classical logic. McCarthy notes, however, that this strategy is much too specialized. It is sometimes

* E-mail: vl@cs.utexas.edu.

SSDI 0004-3702(00013)5

important that only a part of the predicates available in the language be allowed to vary, the others being fixed; otherwise, circumscription may lead to unintuitive consequences.

The general principle seems to be that a predicate should be varied if the purpose of the application of circumscription is to describe the extent of that predicate. McCarthy motivates his decision to vary the predicate *flies* in one of the examples by saying that "the purpose of the axiom set is to describe what flies". Then he writes: "Suppose that we contemplate taking *bird* as variable also. In the first place, this violates the intuition that deciding what flies follows deciding what is a bird in the common-sense situations we want to cover. Secondly, if we use exactly the above axioms and admit bird as variable, we will further conclude that the only birds are penguins, canaries and ostriches." [15, Section 5].

On the other hand, sometimes one needs to vary a function, rather than a predicate. For instance, the situation calculus function *Result* is varied in the solution to the frame problem proposed by Baker [1].

These observations suggest the generalization of simple abnormality theories in which one is allowed to specify, in addition to an axiom set, the predicate and/or function constants $C_1, \ldots, C_m$ that are "described" by the axioms. A possible syntax for such theories is

$$C_1, \ldots, C_m : A_1, \ldots, A_n. \tag{1}$$

The circumscription operator allows us to translate (1) into the language of classical second-order logic by forming the circumscription of the abnormality predicate Ab relative to the conjunction of the axioms $A_1 \wedge \cdots \wedge A_n$ with $C_1, \ldots, C_m$ allowed to vary; symbolically,

$$\mathrm{CIRC}[A_1 \wedge \cdots \wedge A_n; Ab; C_1, \ldots, C_m]. \tag{2}$$

(See [12] for the definition of the circumscription operator and related notation.)

Unfortunately, even this is not general enough. McCarthy describes how to establish priorities among the "aspects" to which the abnormality predicate is applied, and expresses the view "that *prioritized circumscription* will turn out to be the most natural and powerful variant" [15, Section 12]. However, the applications of circumscription to the frame problem in [1] and [11] required yet another generalization—forming a conjunction of several circumscriptions, applied to the same axiom set but having different lists of varied predicates and functions. This was codified in the definition of "circumscriptive theories" proposed in [12, Section 2.6].

The additional flexibility given by these extensions is a mixed blessing. Even after deciding what the axioms will look like, the representer of knowledge still has many choices that may allow him to adjust the circumscription so that its effect will be just right—not too weak and not too strong. The choices are sometimes motivated by a relatively clear principle, such as specificity, but often they have to be made by trial and error. It seems that the difficulty of this process is the main reason why circumscription is not applied today in knowledge representation as widely as could be expected.

In this paper, we propose an alternative approach to the use of circumscription for representing knowledge. Its main idea is to make abnormality theories "nested"—to allow each A_i in (1) to be a "block" of form (1). Intuitively, each block can be viewed as a group of axioms that describes a certain collection of predicates and functions, and the embedding of blocks reflects the dependence of these descriptions on each other. This format allows us to apply the circumscription operator to a subset of the axioms, and not to the whole axiom set, as in [15]. We will see that nested circumscriptions can produce the same results as prioritizations, and often in a more natural way.

The results of "structuring" a knowledge base have been earlier investigated in the context of other nonmonotonic formalisms, including autoepistemic logic [8] and logic programming [9]. Axioms removed from the range of a nonmonotonic interpretation are called "constraints" in [18] and [3], and "observations" in [22].

Another innovation proposed here is the replacement of the predicate constant Ab in (2) by an existentially quantified predicate variable. In formalizations based on circumscription, the abnormality predicate plays an auxiliary role; what we are actually interested in are the logical consequences of (2) that do not include Ab. To put it differently, if (2) is denoted by $F(Ab)$, and ab is a predicate variable of the same arity as Ab, then what we are interested in are the consequences of the sentence $\exists ab F(ab)$. This is a formula not containing Ab, for which (2) is a conservative extension.

For example, the default "Normally $P(x)$" can be expressed by the circumscription

$$\mathrm{CIRC}[\forall x(\neg Ab(x) \supset P(x)); Ab; P],$$

which is equivalent to

$$\forall x \neg Ab(x) \wedge \forall x P(x).$$

We feel that, in this conjunction, the first term is irrelevant, and we would like to "forget" it. Technically, this is achieved by using an existentially quantified predicate variable in place of the predicate constant Ab:

$$\exists ab(\forall x \neg ab(x) \wedge \forall x P(x)).$$

This formula is equivalent to $\forall x P(x)$.

In the context of nested abnormality theories, one effect of this modification is that the abnormality predicate becomes local to the block in which it is used. This is often convenient, and, in many cases, allows us to dispense with "aspects".

2. Blocks and theories

Consider a second-order language L that does *not* include Ab among its symbols. For every natural number k, by L_k we denote the language obtained from L by adding Ab as a k-ary predicate constant. *Blocks* are defined recursively as follows: For any k and any list of function and/or predicate constants[1] $C_1, \ldots, C_m$ $(m \geqslant 0)$ of L, if each of $A_1, \ldots, A_n$ $(n \geqslant 0)$ is a formula of L_k or a *block*, then $\{C_1, \ldots, C_m : A_1, \ldots, A_n\}$ is a *block*. The last expression reads: $C_1, \ldots, C_m$ are such that $A_1, \ldots, A_n$. About $C_1, \ldots, C_m$ we say that they are *described* by this block.

Note that, according to this definition, if A_i and A_j are formulas, and Ab occurs in both, then it is used in both with the same number of arguments; if, however, A_i or A_j is itself a block, then this is not guaranteed.

A *nested abnormality theory* (*NAT*) is a set of blocks, called its *axioms*. Note that each axiom is a finite string of symbols, but there may be infinitely many axioms in a NAT.

The semantics of NATs is characterized by a map φ that translates blocks into sentences of L. It is convenient to make φ defined also on formulas of the languages L_k. If A is such a formula, then φA stands for the universal closure of A. For blocks we define, recursively:

$$\varphi\{C_1, \ldots, C_m : A_1, \ldots, A_n\} = \exists ab F(ab),$$

where

$$F(Ab) = \mathrm{CIRC}[\varphi A_1 \wedge \cdots \wedge \varphi A_n; Ab; C_1, \ldots, C_m].$$

A sentence A of L will be identified with the block $\{: A\}$. It is easy to see that $\varphi\{: A\}$ is equivalent to A.

For any NAT T, φT stands for $\{\varphi A \mid A \in T\}$. Thus φT is a second-order theory in the language L. A *model* of T is a model of φT in the sense of classical logic. A *consequence* of T is a sentence of L that is true in all models of T.

If a block A is an axiom of T, then inserting an additional formula in A may result in losing some of the consequences of T. In this sense, the formalism defined here is nonmonotonic. But adding more axioms to a NAT can only make the set of its consequences larger.

To show how NATs can be used for representing defaults, we will recast several familiar applications of circumscription in the new format.

3. Examples

3.1. Whether birds can fly

As the first illustration, take a standard example: objects normally don't fly; birds normally do; canaries are birds; Tweety is a canary. These assertions can be formalized as the NAT whose only axiom is

[1] This includes function constants of arity 0 (object constants) and predicate constants of arity 0 (propositional constants).

$$
\begin{array}{l}
\{Flies : \\
\quad Flies(x) \supset Ab(x), \\
\quad \{Flies : \\
\qquad Bird(x) \wedge \neg Ab(x) \supset Flies(x), \\
\qquad Canary(x) \supset Bird(x), \\
\qquad Canary(Tweety) \\
\quad \} \\
\}.
\end{array} \tag{3}
$$

The outer block describes the ability of objects to fly; the inner block gives more specific information about the ability of *birds* to fly. This representation of specificity by nesting is different from both methods proposed for this purpose in [15]—cancellation of inheritance axioms and prioritization. Each of these two methods would require the use of aspects (or several abnormality predicates). In a NAT, aspects are only needed when it is important *not* to establish priorities between interacting defaults, because then the defaults have to be placed in the same block. The "Nixon diamond" [21] is an example.

In order to apply φ to (3), we first have to apply φ to the inner block. It is easy to check, using the methods of [12, Section 3], that the result is equivalent to the conjunction of (the universal closures of) the formulas

$$Bird(x) \supset Flies(x), \tag{4}$$

$$Canary(x) \supset Bird(x) \tag{5}$$

and

$$Canary(Tweety). \tag{6}$$

Using this technique again, we conclude that φ applied to (3) is equivalent to the conjunction of (5), (6) and

$$Bird(x) \equiv Flies(x).$$

3.2. Whether canaries are birds

Consider the enhancement of the previous example in which the assertion that all canaries are birds is turned into a default.[2] Now circumscription is to be used for characterizing the extent of the predicate *Bird*, and that calls for introducing one more block. Since "deciding what flies follows deciding what is a bird", the new block will be made innermost:

[2] An inmate who squealed on fellow prisoners would be an exception (John McCarthy, personal conversation).

$$
\begin{array}{l}
\{Flies: \\
\quad Flies(x) \supset Ab(x), \\
\quad \{Flies: \\
\quad\quad Bird(x) \wedge \neg Ab(x) \supset Flies(x), \\
\quad\quad \{Bird: \\
\quad\quad\quad Canary(x) \wedge \neg Ab(x) \supset Bird(x), \\
\quad\quad\quad Canary(Tweety) \\
\quad\quad \} \\
\quad \} \\
\}.
\end{array}
$$

In [15, Section 11], this example is proposed as motivation for introducing priorities. But it is not clear how priorities would help here. The problem is that, in order for the new default to work, one would have to vary *Bird*; that would lead to the undesirable result that there are no birds other than canaries.

3.3. Domain closure assumption

The *domain closure assumption* [19] is the assumption that every object in the universe of discourse is representable by a ground term. The related notion of "domain circumscription" is defined in [13] and reduced to the (now standard) "predicate" circumscription in [14].

The idea of this reduction is to introduce a new kind of atomic formulas, $GT(x)$, expressing that x is representable by a ground term,[3] and postulate

$$\forall x GT(x). \tag{7}$$

The meaning of GT can be expressed by the axioms

$$GT(x_1) \wedge \cdots \wedge GT(x_k) \supset GT(f(x_1, \ldots, x_k)) \tag{8}$$

for each function constant f available in the language, with circumscription used to minimize the extent of GT. (In the special case when f is an object constant, (8) turns into $GT(f)$.)

It is essential here that (7) is added *after* the application of the circumscription operator. Thus circumscription has to be applied to a proper subset of the axioms. This differs from the approach of [15].[4]

It is easy to implement this idea in the framework of NATs. Denote the universal closure of (8) by GT_f, and assume that the language under consideration has finitely many function symbols $f_1, \ldots, f_n$. The domain closure assumption can be expressed by the axioms

[3] McCarthy's notation is $all(x)$.

[4] The fact that, in the early work on circumscription, McCarthy sometimes applied the circumscription operator to a subset of the theory was noted in [4]. Theorem 5.3 in that paper shows that this is, apparently, unavoidable, if the domain closure assumption is to be reduced to circumscription.

$$\begin{aligned}
&\forall x GT(x),\\
&\{GT:\\
&\quad GT(x) \supset Ab(x),\\
&\quad GT_{f_1},\\
&\quad \dots,\\
&\quad GT_{f_n}\\
&\}.
\end{aligned} \tag{9}$$

Let H be the axiom set obtained from (9) by adding the usual unique names axioms (expressing that $f_1, \dots, f_n$ are 1-1 and that their ranges are disjoint). These axioms express that the universe of discourse is isomorphic to the Herbrand universe of the language—to the set of its ground terms. For any NAT T, the models of $T \cup H$ are the "Herbrand models" of T.

4. Two special cases

4.1. Minimizing a predicate

We will abbreviate blocks of the form

$$\{P : P(x) \supset Ab(x), A_1, \dots, A_n\},$$

where P is a predicate constant and x a tuple of distinct variables, as

$$\{\min P : A_1, \dots, A_n\}.$$

For instance, the second axiom in (9) can be written as

$$\begin{aligned}
&\{\min GT:\\
&\quad GT_{f_1},\\
&\quad \dots,\\
&\quad GT_{f_n}\\
&\}.
\end{aligned}$$

More generally, we will write

$$\{C_1, \dots, C_m, P : P(x) \supset Ab(x), A_1, \dots, A_n\}$$

as

$$\{C_1, \dots, C_m, \min P : A_1, \dots, A_n\}. \tag{10}$$

If $A_1, \dots, A_n$ are sentences (rather than blocks), and Ab does not occur in any of them, then (10) has the same meaning as the circumscription of P:

Proposition 1. *If $A_1, \dots, A_n$ are sentences that do not contain Ab, then*

$$\varphi\{C_1, \dots, C_m, \min P : A_1, \dots, A_n\} \tag{11}$$

is equivalent to

$$\mathrm{CIRC}[A_1 \wedge \cdots \wedge A_n; P; C_1, \ldots, C_m]. \tag{12}$$

Proof. Denote the conjunction $A_1 \wedge \cdots \wedge A_n$ by $A(P, C)$, where C stands for the list $C_1, \ldots, C_m$. Then (11) can be written as

$$\exists ab[P \leqslant ab \wedge A(P, C) \wedge \neg\exists ab' pc(ab' < ab \wedge p \leqslant ab' \wedge A(p, c))]. \tag{13}$$

The third conjunctive term in the brackets can be simplified as follows:

$$\begin{aligned}&\neg\exists ab' pc(ab' < ab \wedge p \leqslant ab' \wedge A(p, c))\\ &\quad \equiv \neg\exists pc(\exists ab'(p \leqslant ab' \wedge ab' < ab) \wedge A(p, c))\\ &\quad \equiv \neg\exists pc(p < ab \wedge A(p, c)).\end{aligned}$$

This formula implies

$$\neg(P < ab \wedge A(P, C)).$$

In combination with the first two conjunctive terms in (13), the last formula implies $P = ab$. Consequently, (13) is equivalent to

$$\exists ab[P = ab \wedge A(P, C) \wedge \neg\exists pc(p < ab \wedge A(p, c))].$$

This can be further rewritten as

$$A(P, C) \wedge \neg\exists pc(p < P \wedge A(p, c)),$$

which is identical to (12). $\square$

4.2. *Maximizing a predicate*

It is convenient to write a block of the form

$$\{C_1, \ldots, C_m, P : \neg Ab(x) \supset P(x), A_1, \ldots, A_n\}$$

as

$$\{C_1, \ldots, C_m, \max P : A_1, \ldots, A_n\}.$$

Maximizing a predicate is equivalent to minimizing its negation:

Proposition 2. *Let $A_1(P), \ldots, A_n(P)$ be formulas, and let $\overline{P}$ be a predicate constant that does not occur in any of them. If $F(P)$ is the sentence*

$$\varphi\{C_1, \ldots, C_m, \max P : A_1(P), \ldots, A_n(P)\}$$

and $G(\overline{P})$ is the sentence

$$\varphi\{C_1, \ldots, C_m, \min \overline{P} : A_1(\lambda x \neg\overline{P}(x)), \ldots, A_n(\lambda x \neg\overline{P}(x))\},$$

then $F(P)$ is equivalent to $G(\lambda x \neg P(x))$.

Proof. Denote the conjunction of the universal closures of $A_1(P),\ldots,A_n(P)$ by $A(Ab,C,P)$, where C stands for the list $C_1,\ldots,C_m$. Then $F(P)$ is

$$\begin{aligned}&\exists ab[\lambda x\neg P(x) \leqslant ab \wedge A(ab,P,C)\\&\quad\wedge \neg\exists ab' pc(ab' < ab \wedge \lambda x\neg p(x) \leqslant ab' \wedge A(ab',p,c))],\end{aligned}$$

which is equivalent to

$$\begin{aligned}&\exists ab[\lambda x\neg P(x) \leqslant ab \wedge A(ab,P,C)\\&\quad\wedge \neg\exists ab' c(ab' < ab \wedge \exists p(\lambda x\neg p(x) \leqslant ab' \wedge A(ab',p,c)))].\end{aligned} \tag{14}$$

Furthermore, $G(\overline{P})$ is

$$\begin{aligned}&\exists ab[\overline{P} \leqslant ab \wedge A(ab,\lambda x\neg\overline{P}(x),C)\\&\quad\wedge \neg\exists ab'\overline{p}c(ab' < ab \wedge \overline{p} \leqslant ab' \wedge A(ab',\lambda x\neg\overline{p}(x),c))],\end{aligned}$$

so that $G(\lambda x\neg P(x))$ is equivalent to

$$\begin{aligned}&\exists ab[\lambda x\neg P(x) \leqslant ab \wedge A(ab,P,C)\\&\quad\wedge \neg\exists ab'\overline{p}c(ab' < ab \wedge \overline{p} \leqslant ab' \wedge A(ab',\lambda x\neg\overline{p}(x),c))].\end{aligned}$$

This formula can be further rewritten as

$$\begin{aligned}&\exists ab[\lambda x\neg P(x) \leqslant ab \wedge A(ab,P,C)\\&\quad\wedge \neg\exists ab' c(ab' < ab \wedge \exists\overline{p}(\overline{p} \leqslant ab' \wedge A(ab',\lambda x\neg\overline{p}(x),c)))].\end{aligned}$$

In order to show that it is equivalent to (14), it suffices to notice that

$$\exists\overline{p}(\overline{p} \leqslant ab' \wedge A(ab',\lambda x\neg\overline{p}(x),c))$$

is equivalent to

$$\exists p(\lambda x\neg p(x) \leqslant ab' \wedge A(ab',p,c)). \qquad \square$$

5. More examples: the frame problem

5.1. Causal minimization

Consider now the formalization of the Yale Shooting example that uses the "causal minimization" method [10]. It involves variables for actions (a), for situations (s), and for truth-valued fluents (f). The object constants are: actions *Load*, *Wait* and *Shoot*; situation *S*0; fluents *Loaded* and *Alive*. The binary situation-valued function *Result* has an action term and a situation term as arguments. There are four predicate constants: $Holds(f,s)$ expresses that f is true in situation s; $Precondition(f,a)$ expresses that f is a precondition for the execution of a; $Causes^+(a,f)$ expresses that a causes f to become true; $Causes^-(a,f)$ expresses that a causes f to become false.[5] Abbreviations: $Succeeds(a,s)$ stands for

$$\forall f(Precondition(f,a) \supset Holds(f,s));$$

[5] In [10], a ternary predicate *Causes* is used in place of two binary predicates $Causes^+$, $Causes^-$.

Affects(a, f, s) stands for

$$Succeeds(a, s) \wedge (Causes^{+}(a, f) \vee Causes^{-}(a, f)).$$

One group of axioms characterizes the predicates *Precondition*, *Causes*$^{+}$ and *Causes*$^{-}$:

$$\begin{aligned}
&Precondition(Loaded, Shoot),\\
&Causes^{+}(Load, Loaded),\\
&Causes^{-}(Shoot, Loaded),\\
&Causes^{-}(Shoot, Alive).
\end{aligned} \tag{15}$$

The main idea of [10] is to use circumscription to guarantee that these predicates are true only when this is required by axioms (15)—to force these predicates to satisfy the "closed world assumption" relative to these axioms. The closed world assumption for *Precondition* implies that actions have no unintended preconditions. This solves the qualification problem. The closed world assumption for *Causes*$^{+}$ and *Causes*$^{-}$ implies that no unintended changes take place in the world when an action is performed. This solves the frame problem.

It is not easy, however, to implement this plan. Merely circumscribing the three predicates would not enforce the closed world assumption relative to axioms (15), because the predicates occur in other axioms also (by virtue of being used in the abbreviations *Succeeds* and *Affects*). A part of the solution is to carefully select the circumscription policy, and to allow *Holds* to vary when *Precondition*, *Causes*$^{+}$ and *Causes*$^{-}$ are circumscribed. This achieves the goal at least if we restrict attention to "term models" of the circumscription [10, Section 3]. This restriction can be discarded at the price of making the language and the axioms more complicated [10, Proposition 2].

These difficulties would not arise in the framework of NATs. In order to express that a subset of axioms is a complete definition of some predicate, we simply turn this subset into a block.

Before presenting the causal minimization method in terms of NATs, we need to extend the syntax and semantics of NATs to the case when the underlying language L is many-sorted. The only place in Section 2 above that needs to be modified is the definition of L_k. In a many-sorted language, a predicate symbol is characterized not only by its arity, but also by the sort σ_1 of its first argument, the sort σ_2 of its second argument, etc. Accordingly, instead of understanding k as a natural number, we should allow k to be a finite string $\sigma_1\sigma_2\ldots$ in the alphabet whose characters correspond to the sorts of L.

Now we are ready to describe the causal minimization treatment of Yale Shooting as a NAT.

(1) *Unique names axioms*:

$$\begin{aligned}
&Load \neq Wait,\\
&Load \neq Shoot,\\
&Wait \neq Shoot,\\
&Loaded \neq Alive.
\end{aligned}$$

(2) *Initial conditions*:

$$Holds(Alive, S0),$$
$$\neg Holds(Loaded, S0).$$

(3) *Definition of Precondition*:

$$\{\min Precondition : \quad Precondition(Loaded, Shoot) \quad \}.$$

(4) *Definitions of $Causes^+$ and $Causes^-$*:

$$\{\min Causes^+ : \quad Causes^+(Load, Loaded) \quad \},$$
$$\{\min Causes^- : \quad Causes^-(Shoot, Loaded), \quad Causes^-(Shoot, Alive) \quad \}.$$

(5) *General laws of motion*:

$$Succeeds(a, s) \wedge Causes^+(a, f) \supset Holds(f, Result(a, s)),$$
$$Succeeds(a, s) \wedge Causes^-(a, f) \supset \neg Holds(f, Result(a, s)),$$
$$\neg Affects(a, f, s) \supset (Holds(f, Result(a, s)) \equiv Holds(f, s)).$$

Using Proposition 1 and the methods of [12, Section 3], we can rewrite the axioms of Groups 3 and 4 as explicit definitions of *Precondition*, $Causes^+$ and $Causes^-$ in the sense of first-order logic. Then we can easily verify, using the axioms of Group 1, that all ground instances of $Causes^+(a, f)$, $Causes^-(a, f)$ and $Precondition(f, a)$ are decidable, and that the universal quantifier in the definition of *Succeeds* can be replaced by a finite conjunction. It follows by induction that all ground instances of $Holds(f, s)$ are decidable also. In [10], the corresponding completeness result was rather difficult; in the new framework, it becomes quite transparent.

5.2. Baker's method

The circumscriptive solution to the frame problem that uses the existence of situation axioms [1] is reformulated as a NAT in [7]. Here again, the formulation is simpler in the new framework, and the effect of circumscription is easier to investigate. Moreover, the formulation in terms of NATs is applicable to nondeterministic actions. Kartha [6] showed that this is not the case for Baker's original solution.

5.3. Explanation closure

Another approach to the frame problem, developed by Haas [5], Schubert [23,24] and Reiter [20], is based on the idea of "explanation closure". The process of generating explanation closure axioms can be conveniently described in terms of NATs. We will illustrate this fact with an example borrowed from [20].

The language has variables for robots (r), for the objects that they handle (b, x), for actions (a) and for situations (s). The explanation closure method will be used to describe how the property of being broken—symbolically, $Broken(x,s)$—is affected by actions of three types: by dropping x on the floor, by exploding a bomb near x, and by repairing x. To this end, two auxiliary predicates are introduced, $Broken^+$ and $Broken^-$. The formula $Broken^+(x,s,a)$ expresses that a changes the value of $Broken(x,s)$ to *true*. Similarly, $Broken^-(x,s,a)$ says that a changes the value of $Broken(x,s)$ to *false*. The possibility of doing a in situation s is expressed by $Poss(a,s)$.

(1) *Unique names axioms*:

$$\begin{aligned}
&Drop(r_1,x_1) = Drop(r_2,x_2) \supset (r_1 = r_2 \wedge x_1 = x_2),\\
&Repair(r_1,x_1) = Repair(r_2,x_2) \supset (r_1 = r_2 \wedge x_1 = x_2),\\
&Explode(b_1) = Explode(b_2) \supset b_1 = b_2,\\
&Drop(r_1,x_1) \neq Repair(r_2,x_2),\\
&Drop(r,x) \neq Explode(b),\\
&Repair(r,x) \neq Explode(b).
\end{aligned}$$

(2) *Definition of Poss*:

$$\begin{aligned}
&\{\max Poss :\\
&\quad Poss(Drop(r,x),s) \supset Holding(r,x,s),\\
&\quad Poss(Repair(r,x),s) \supset HasGlue(r,s),\\
&\quad Poss(Repair(r,x),s) \supset Broken(x,s),\\
&\quad Poss(Explode(b),s) \supset Bomb(b)\\
&\}.
\end{aligned}$$

(3) *Definitions of* $Broken^+$ *and* $Broken^-$:

$$\begin{aligned}
&\{\min Broken^+ :\\
&\quad Fragile(x) \supset Broken^+(x,s,Drop(r,x)),\\
&\quad NextTo(b,x,s) \supset Broken^+(x,s,Explode(b))\\
&\},\\
&\{\min Broken^- :\\
&\quad Broken^-(x,s,Repair(r,x))\\
&\}.
\end{aligned}$$

(4) *Effect axioms*:

$$Poss(a,s) \wedge Broken^{+}(x,s,a) \supset Broken(x, Result(a,s)),$$
$$Poss(a,s) \wedge Broken^{-}(x,s,a) \supset \neg Broken(x, Result(a,s)).$$

(5) *Explanation closure axioms*:

$$Poss(a,s) \wedge \neg Broken(x,s) \wedge Broken(x, Result(a,s)) \supset Broken^{+}(x,s,a),$$
$$Poss(a,s) \wedge Broken(x,s) \wedge \neg Broken(x, Result(a,s)) \supset Broken^{-}(x,s,a).$$

Using Propositions 1 and 2 and the methods of [12, Section 3], we can rewrite the axioms of Groups 2 and 3 as explicit definitions of *Poss*, $Broken^{+}$ and $Broken^{-}$ in the sense of first-order logic. Having replaced $Broken^{+}$ and $Broken^{-}$ by their definitions in the axioms of Groups 4 and 5, we will arrive at the formulation of effect axioms and explanation closure axioms identical to the one given in [20, Section 3.1].

There is a striking similarity between the two solutions to the frame problem—causal minimization and explanation closure—when each is presented as a NAT. The main difference is that the latter does not use fluent variables. This simplicity comes at a price, however: without fluent variables, explanation closure axioms for different fluents cannot be combined into a small number of general axioms, such as the "general laws of motion" in the causal minimization method.

Schubert [24], whose description of the explanation closure method does not appeal to circumscription, argues that the success of the method calls for "a reassessment of the proper roles" of monotonic and nonmonotonic approaches to reasoning about action. From our perspective, stressing the difference between the explanation closure approach as "monotonic" and the others as "nonmonotonic" is not fully justified. Since circumscription is merely a syntactic transformation of formulas, any circumscriptive representation of a body of knowledge can be viewed as an abbreviated form of a representation in classical logic. Circumscriptive representations are attractive when they are more compact and manageable than the formalizations that use classical logic directly. The example above suggests that the explanation closure method may be in this category.

Reiter [20] generates first-order explanation closure axioms from effect axioms using a process similar to Clark's completion [2]. The circumscriptive presentation of explanation closure may lead to a generalization of this method that will be applicable to nondeterministic actions. [6]

6. Conclusion

The concept of a nested abnormality theory serves as a basis for a new style of applying circumscription to representing defaults. Sometimes it permits more economical and elegant formulations than the traditional ones, based on simple or prioritized circumscription. Sometimes it leads to satisfactory solutions where prioritized circumscription

[6] This suggestion was made by Raymond Reiter when this work was presented at the 1994 Festival of Action Formalisms at the University of Toronto.

seems to fail. The effect of circumscription in a nested abnormality theory is often easier to compute. These advantages are due to the fact that, in a nested abnormality theory, the circumscription operator can be applied to a small subset of axioms.

One attractive feature of nonmonotonic formalizations of knowledge is that they are often "elaboration tolerant" [16] to a larger degree than formalizations based on classical logic. It is often possible to enhance a nonmonotonic theory by simply adding new formulas to the axiom set, whereas the corresponding enhancement of a classical axiomatization would require changing the existing axioms. This happens, for instance, when we want to enhance a description of an action domain by postulating additional effects of actions, or by assuming new preconditions. Introducing a block structure in the axiom set clearly limits the degree of elaboration tolerance that can be achieved, and one may ask whether the proposal presented in this paper defeats the very purpose of the nonmonotonic enterprise.

This is a serious criticism. We would have preferred to use traditional "one-level", or "unstructured" axiom sets, if that did not prevent circumscription from becoming a convenient knowledge representation tool. Unfortunately, the one-level approach does not seem to be successful. But it appears that, even with nesting, circumscription leads to a useful form of elaboration tolerance if each block represents an intuitively meaningful structural unit, a reasonable "group of axioms". This is the case, for instance, in our formulations of causal minimization and explanation closure.

Acknowledgements

I would like to thank Michael Gelfond, Raymond Reiter and the anonymous referees for useful comments. This work was supported in part by National Science Foundation under grant IRI-9306751.

References

[1] A. Baker, Nonmonotonic reasoning in the framework of situation calculus, *Artif. Intell.* **49** (1991) 5-23.
[2] K. Clark, Negation as failure, in: H. Gallaire and J. Minker, eds., *Logic and Data Bases* (Plenum Press, New York, 1978) 293-322.
[3] L. Console, D.T. Dupre and P. Torasso, On the relationship between abduction and deduction, *J. Logic Comput.* **1** (5) (1991) 661-690.
[4] D. Etherington, R. Mercer and R. Reiter, On the adequacy of predicate circumscription for closed-world reasoning, *Comput. Intell.* **1** (1) (1985).
[5] A. Haas, The case for domain-specific frame axioms, in: F.M. Brown, ed., *The Frame Problem in Artificial Intelligence, Proceedings of the 1987 Workshop* (1987).
[6] G.N. Kartha, Two counterexamples related to Baker's approach to the frame problem, *Artif. Intell.* **69** (1994) 379-391.
[7] G.N. Kartha and V. Lifschitz, Actions with indirect effects (preliminary report), in: *Proceedings Fourth International Conference on Principles of Knowledge Representation and Reasoning*, Bonn, Germany (1994) 341-350.
[8] K. Konolige, Hierarchic autoepistemic theories for non-monotonic reasoning, in: M. Reinfrank, J. de Kleer, M. Ginsberg and E. Sandewall, eds., *Non-Monotonic Reasoning: Second International Workshop*, Lecture Notes in Artificial Intelligence **346** (Springer-Verlag, Berlin, 1989).

[9] E. Laenenes and D. Vermeir, A fixpoint semantics for ordered logic, *J. Logic Comput.* **1** (2) (1990) 159-186.

[10] V. Lifschitz, Formal theories of action, in: F.M. Brown, ed., *The Frame Problem in Artificial Intelligence, Proceedings of the 1987 Workshop* (1987) 35-58.

[11] V. Lifschitz, Towards a metatheory of action, in: J. Allen, R. Fikes and E. Sandewall, eds., *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, Cambridge, MA (1991) 376-386.

[12] V. Lifschitz, Circumscription, in: D.M. Gabbay, C.J. Hogger and J.A. Robinson, eds., *The Handbook of Logic in AI and Logic Programming* **3** (Oxford University Press, Oxford, 1993) 298-352.

[13] J. McCarthy, Epistemological problems of Artificial Intelligence, in: *Proceedings IJCAI-77*, Cambridge, MA (1977) 1038-1044. Reproduced in: J. McCarthy, *Formalizing Common Sense: Papers by John McCarthy* (Ablex, Norwood, NJ, 1990).

[14] J. McCarthy, Circumscription—a form of non-monotonic reasoning, *Artif. Intell.* **13** (1-2) (1980) 27-39, 171-172. Reproduced in: J. McCarthy, *Formalizing Common Sense: Papers by John McCarthy* (Ablex, Norwood, NJ, 1990).

[15] J. McCarthy, Applications of circumscription to formalizing common sense knowledge, *Artif. Intell.* **26** (3) (1986) 89-116. Reproduced in: J. McCarthy, *Formalizing Common Sense: Papers by John McCarthy* (Ablex, Norwood, NJ, 1990).

[16] J. McCarthy, Mathematical logic in Artificial Intelligence, *Dædalus* (1988) 297-311. Reproduced in: J. McCarthy, *Formalizing Common Sense: Papers by John McCarthy* (Ablex, Norwood, NJ, 1990).

[17] J. McCarthy, *Formalizing Common Sense: Papers by John McCarthy* (Ablex, Norwood, NJ, 1990).

[18] D. Poole, A logical framework for default reasoning, *Artif. Intell.* **36** (1) (1988) 27-48.

[19] R. Reiter, Equality and domain closure in first-order data bases. *J. ACM* **27** (2) (1980) 235-249.

[20] R. Reiter, The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression, in: V. Lifschitz, ed., *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy* (Academic Press, New York, 1991) 359-380.

[21] R. Reiter and G. Criscuolo, On interacting defaults, in: *Proceedings IJCAI-81*, Vancouver, BC (1981) 270-276.

[22] E. Sandewall, Filter preferential entailment for the logic of action in almost continuous worlds, in: *Proceedings IJCAI-89*, Detroit, MI (1989) 894-899.

[23] L. Schubert, Monotonic solution of the frame problem in the situation calculus: an efficient method for worlds with fully specified actions, in: H.E. Kyburg, R. Loui and G. Carlson, eds., *Knowledge Representation and Defeasible Reasoning* (Kluwer, Dordrecht, 1990) 23-67.

[24] L. Schubert, Explanation closure, action closure, and the Sandewall test suite for reasoning about change, Tech. Report 440, University of Rochester, Rochester, NY (1992).